Høgskolen i Buskerud

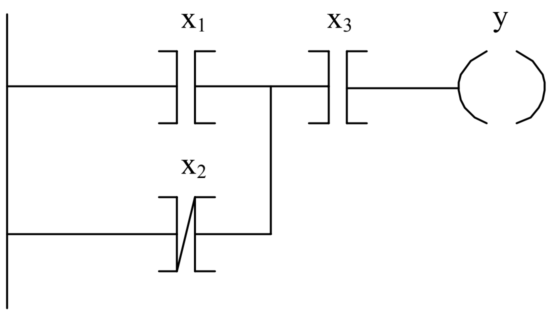
Løsningsforslag til slutteksamen i SESM3401 Styring av mekatroniske systemer

Utarbeidet av Finn Haugen, emnets lærer.

Eksamensdato: Mandag 11. desember 2006. Varighet: 4 timer. Vekt i sluttkarakteren: 40%. Hjelpemidler: Ingen trykte eller håndskrevne hjelpemidler. Kalkulator ikke tillatt.

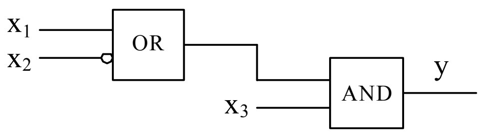
---

1. (15% vekt i dette oppgavesettet) Figur 1 viser ladderdiagrammet.



Figur 1: Ladderdiagram

Figur 2 viser funksjonsblokkdiagrammet.



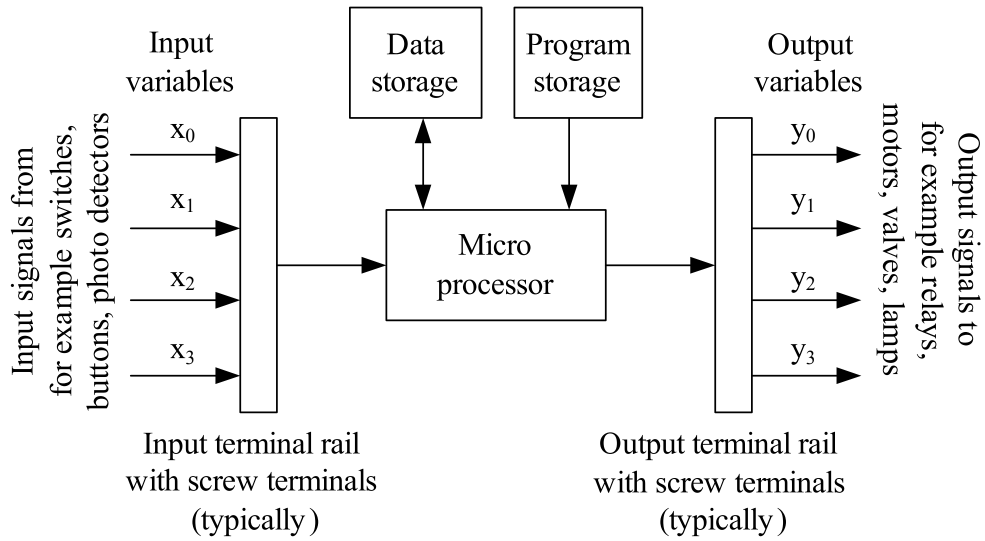
Figur 2: Funksjonsblokkdiagram

2. (15%) Anta at Stopp = Off. Da er $\overline{\text{Stopp}}$ = On, og den tilhørende NC-kontakten (normally closed) er lukket. Anta videre at Motor = Off, og at Start blir On. Da vil Motor bli On (motoren starter). Start blir så Off, men siden Motor-signalet inngår i en OR-kombinasjon, vil Motor forbli On (dette er selvhold). Anta så at Stopp blir On. Da vil $\overline{\text{Stopp}}$ bli Off, og motoren stopper, og Motor vil forbli Off selv når Stopp blir Off.

3. (15%) (Litt engelsk gjør vel ikke noe :-) PLC is short for Programmable Logical Controller. PLC-systems are modular systems

for logical (binary) and sequential control of valves, motors, lamps etc. Modern PLCs includes function modules for PID control. The programming is usually executed on a PC, and the PLC-program is then downloaded (transfered) to the CPU in the PLC-system which then can be disconnected from the PC. The program languages are standardized (the IEC 61131-3 standard), but the actual languages which are implemented on commercial PLCs may differ somewhat from the standard. PLC programming is described in more detail in Section **??**.

Figure 3 shows the main parts of a PLC. The PLC operates as



Figur 3: PLS

follows: The control program executes the program code which is stored in the program storage (memory). The executable program is typically developed on a PC and then downloaded from the PC to the PLC. The data storage contains data uses by the program. Typically the program uses data from the physical inputs, together with data in the data storage, to calculate output values. The program may execute periodically. The cycle time or period is typically in the range of 10ms. The program may also execute if a certain event occurs, e.g. if a program interrupt signal is generated by a digital (boolean) input or by some other external or internal signal source.

4. (10%) PACs constitute an alternative to PLCs as control hardware. One example of a PAC is National Instruments' (Compact) FieldPoint. It is here assumed that the FieldPoint rack contains a RT

unit (Real-time) which contains a microprocessor running a real-time operating system which can run LabVIEW programs downloaded from a PC where the program was developed. To download LabVIEW code, the LabVIEW Real-Time Module must be installed on the PC. Once downloaded, the PAC can be run independently of the PC. National Instruments denotes this equipent a *PAC* – Programmable Automation Controller. The PAC is a modular system similar to PLCs in several aspects. Logical and sequential control and PID control can be realized in the PAC. In fact, all kinds of applications can be developed with LabVIEW and downloaded to the PAC.

5. (15%) Gitt PI-regulatoren

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e\left(\tau\right) d\tau \tag{1}$$

Derivering av begge sider gir

$$\dot{u}(t) = K_p \dot{e}(t) + \frac{K_p}{T_i} e\left(t\right) \tag{2}$$

som, med bruk av Eulers bakoverapproksimasjon for de deriverte, gir

$$\frac{u(t_k) - u(t_{k-1})}{h} = K_p \frac{e(t_k) - e(t_{k-1})}{h} + \frac{K_p}{T_i} e\left(t_k\right) \tag{3}$$

som løst mhp. $u(t_k)$ gir

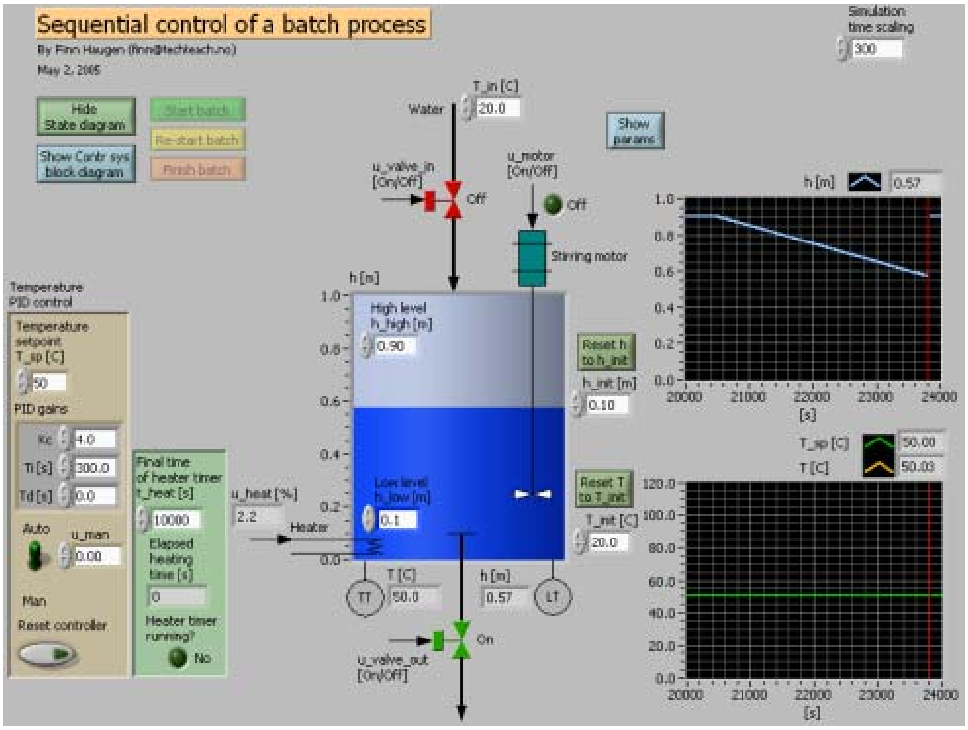
$$u(t_k) = u(t_{k-1}) + K_p e(t_k) - e(t_{k-1}) + \frac{K_p h}{T_i} e\left(t_k\right) \tag{4}$$

6. (20%) Eksempel: Figur 4 viser en batch-prosess. (Denne prosessen er gjennomgått i undervisningen.) Figur 5 viser et SFC-diagram som implementerer sekvensiell styring av prosessen.

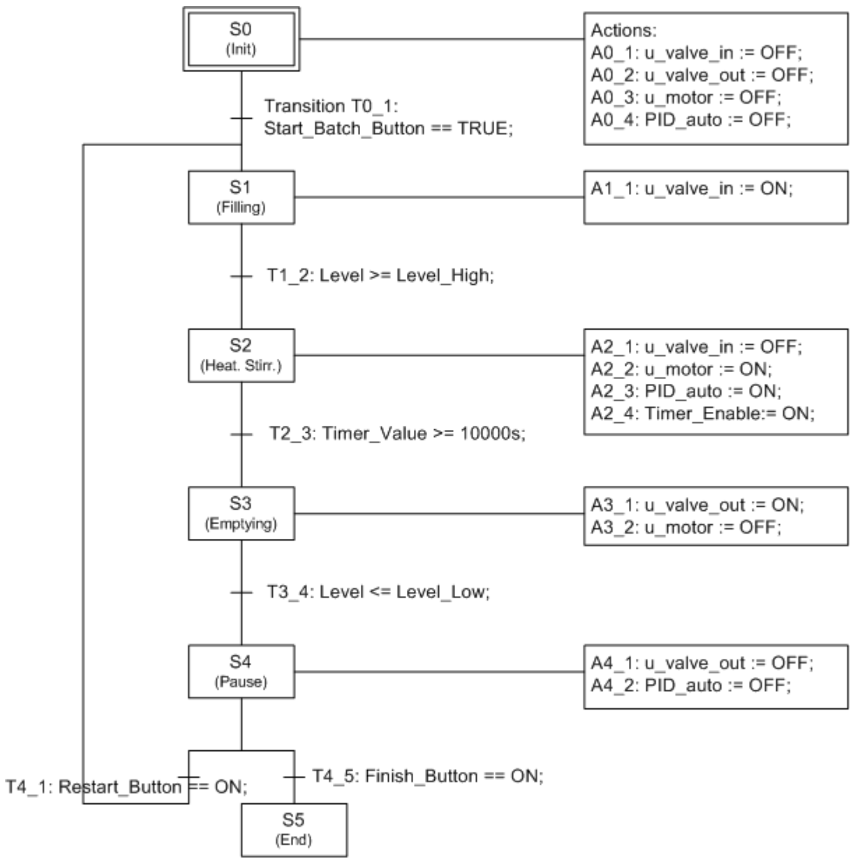
7. (10%) Hardware-in-the-loop-simulering eller HIL-simulering er å bruke det aktuelle automatiseringssutstyret (f.eks. en industriell PID-regulator eller en PLS eller en prototype av en egenutviklet styringsenhet) til å regulere eller styre en simulert prosess. Simulatoren må da kjøre i sann tid eller i skalert sann tid. En slik simulator er gjerne implementert i en PC. PLS'en og simulatoren kommuniserer via ordinær I/O, dvs. med de ordinære styresignalene og målesignalene (strøm og spenning). Aktuelle simuleringsverktøy er LabVIEW Simulation Module og Simulink.

   Hensikten med HIL-simulering kan være uttesting eller opplæring.

   Et konkret eksempel på HIL-simulering: En Mitsubishi-PLS styrer en simulert DC-motor. Motorens simulator er implementert i LabVIEW Simulation Module. I/O med PC'en skjer med I/O-enheten USB-6009.



Figur 4: Batch-prosess som skal styres med SFC



Figur 5: SFC-diagram